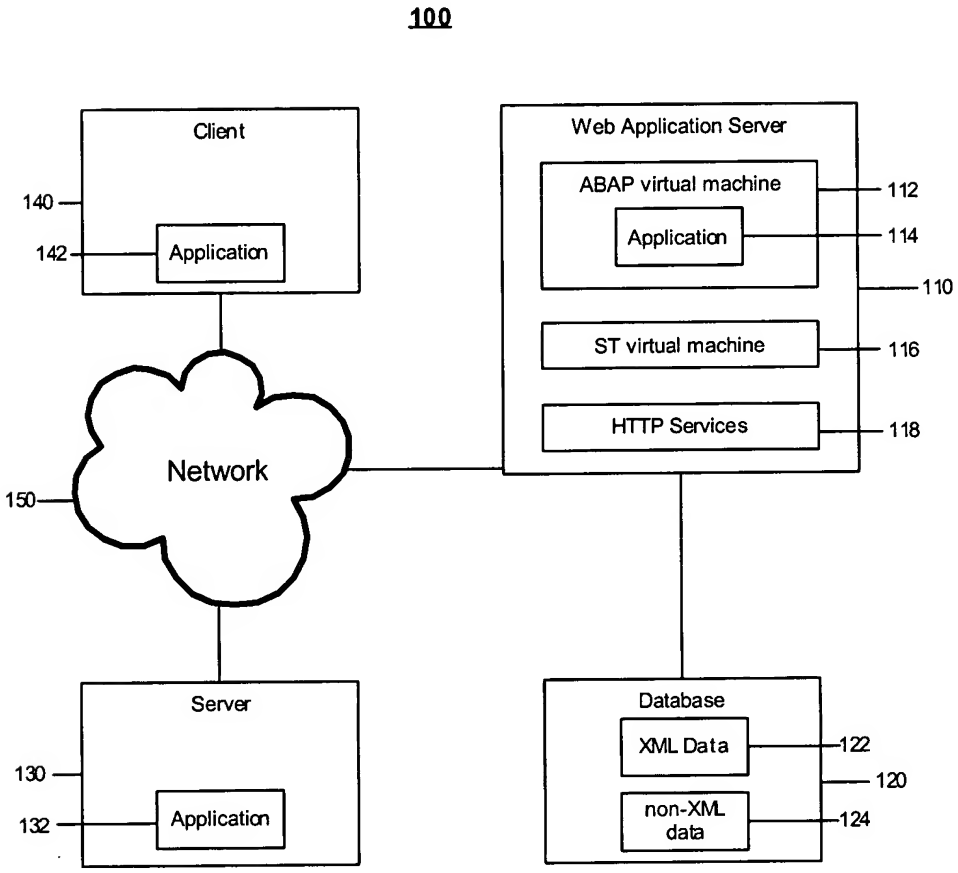
100
Client
140
142
Application
Web Application Server
ABAP virtual machine
Application
112
114
110
ST virtual machine
116
HTTP Services
118
Network
150
Server
130
132
Application
Database
XML Data
122
120
non-XML data
124

FIG. 1

200

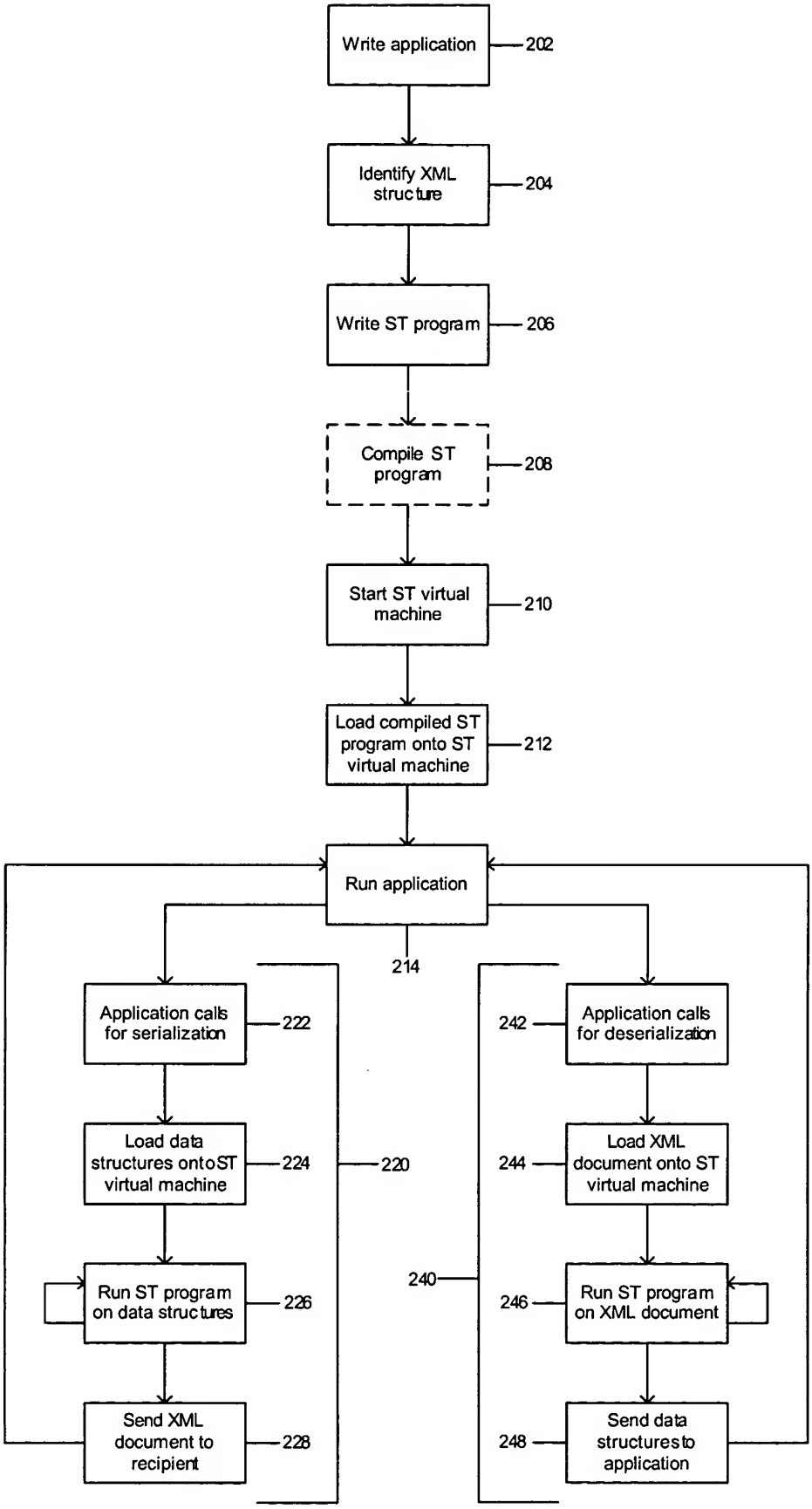
Write application
202
Identify XML structure
204
Write ST program
206
Compile ST program
208
Start ST virtual machine
210
Load compiled ST program onto ST virtual machine
212
Run application
214
Application calls for serialization
222
Load data structures onto ST virtual machine
224
220
Run ST program on data structures
226
Send XML document to recipient
228
242
Application calls for deserialization
Load XML document onto ST virtual machine
244
240
Run ST program on XML document
246
Send data structures to application
248

FIG. 2

300

```
<tt:transform version="0.1"                                          302
 xmlns:tt="http://www.sap.com/transformation-templates"              302
 xmlns:p1="U1"                                                       302
>                                                                    302

<!-- ******************************************************          398
  SAP Simple Transformations sample program, demonstrates            398
  most ST instructions                                               398
  Copyright (c) 2003, SAP AG, Germany. All rights reserved.          398
 ****************************************************** -->         398

<tt:include name="include.xtt"/>                                     304

<tt:root name="R1"/>                                                 306
<tt:root name="R2"/>                                                 308

<tt:template>                                                        310

  <x0>                                                               312

<!-- literal element with literal attributes -->                     398
    <x1 a11="v11" a12="v12">                                         314
<!-- ref-node value -->                                              398
      <tt:value ref="R1.C1"/>                                        316
    </x1>                                                            318

<!-- element with ref-node value -->                                 398
    <x2 tt:value-ref="R1.C2"/>                                       320

<!-- element with literal value and lax matching -->                 398
    <x3 tt:lax="y"><tt:text> text3</tt:text></x3>                    322

<!-- element in namespace -->                                        398
    <p1:x4 xmlns:p2="U2" tt:ref="R2.C1">                             324
<!-- attribute with ref-node value -->                               398
      <tt:attribute name="a41" value-ref=".R1.C3"/>                  326
<!-- attribute with complex content -->                              398
      <tt:attribute name="p2:a42" ref=".R1.C4">                      328
        <tt:text>text42</tt:text>                                    330
        <tt:value/>                                                  332
      </tt:attribute>                                                334
<!-- empty element -->                                               398
      <x41/>                                                         336
<!-- value with follow-text -->                                      398
      <tt:value/>
      <tt:text>text4</tt:text>
    </p1:x4>

<!-- value with special mappings -->                                 398
    <tt:value ref="R2.C2"
     map=" s('~','+','-'), d('*','~') "/>

<!-- copy of sub-tree -->                                            398
    <tt:copy ref="R2.C3"/>                                           346

<!-- loop over table -->                                             398
    <tt:loop ref="R2.C4"><x5 tt:value-ref="LC1"/></tt:loop>          348

<!-- skip elements -->                                               398
    <tt:skip name="p1:x6" count="*"/>                                350

<!-- call -->                                                        398
    <tt:call transformation="ST1">                                   352
      <tt:with-root name="RR1" ref="R2.C5"/>                         354
    </tt:call>                                                       356

<!-- basic conditional with assertion -->                            398
    <tt:cond data="equal('R2.C7',42)">                               358
      <x7 tt:value-ref="R2.C8"/>                                     360
    </tt:cond>                                                       362

<!-- switch -->                                                      398
    <x8 tt:ref="R1">                                                 364
      <tt:switch>                                                    366
        <tt:cond data="initial(C5)">                                 368
          <tt:attribute name="nil">true</tt:attribute>               370
        </tt:cond>                                                   372
        <tt:s-cond data="check(less(C5,100))">99</tt:s-cond>         374
        <tt:cond data="check(less-equal(C5,200) and                  376
         (not(type-I(C4)) or greater-ref(C5,C4)))">                  376
          <tt:value ref="C5"/>                                       378
        </tt:cond>                                                   380
        <tt:d-cond data="equal(C5,100)">                             382
          <tt:empty/>                                                384
        </tt:d-cond>                                                 386
      </tt:switch>                                                   388
    </x8>                                                            390

  </x0>                                                              392

</tt:template>                                                       394

</tt:transform>                                                      396
```

400

FIG. 3